Page 13: Amend the first full paragraph starting at line 5 as follows:

In addition, the present invention is a processor having a memory for storing an instruction code, an instruction code holding means for holding the instruction code read from said memory, and a plurality of computing units constituted by at least one computing device controlled by information held in said instruction code holding means and a register file for storing operand information of said computing device, wherein said instruction code is constituted by a plurality <u>of</u> fields corresponding to the number of computing units, at least one computing device having a same function is provided in all of said computing units and a special register for holding a data type having a bit width too large to specify by a register in said register file is provided in each of said computing units, thereby allowing computational processing of both of a data type having a bit width specifiable by a register in said register file and the data type stored in said special data type.

Page 27: Amend the paragraph starting at line 20 and ending at page 29, line 6, as follows:

In what follows, the present invention will be described. FIG. 1 is a block diagram illustrating a VLIW processor to which the present invention is applied. In the figure, reference numeral 1 denotes an instruction memory for storing a compressed instruction, reference numeral 2 denotes an instruction expansion circuit, a main block of the present invention, for expanding a compressed instruction code read from the instruction memory 1 into an actually executable code, reference numeral 3 denotes an address bus of the instruction memory 1, reference numeral 4 denotes a data bus of the instruction memory 1, reference numerals 5 through 12 denote field buses to which the instruction expansion circuit 2 ~~output~~ <u>outputs</u> an expanded code, reference numerals 14 through 21 denote instruction registers for holding expanded codes transferred via the field buses 5 through 12, reference numerals 22 through 25 denote computing ~~unit~~ <u>units</u> having a same constitution for executing various computational operations according to the expanded codes held in the instruction registers 14 through 21, reference numeral 26 denotes an IFG (Integer Floating Graphics) computing device for executing complicated computational operations such as a multimedia

computation operation for which a plurality of operations are performed on an 8-bit or 16-bit basis and a multiplication, reference numeral 27 denotes an INT (Integer) computing device for executing simple computational operations such as a data transfer instruction for executing data transfer between a data memory 30 and a register file and a logic operation, reference numeral 28 denotes a register file for holding a value to be operated and an operation result value, composed of 32 64-bit registers, and having 4 read ports and 3 write ports, reference numeral 29 denotes a selection circuit for transferring operation results of the computing units 22 through 25 to another operation unit, and reference numeral 30 denotes the data memory with which data is transferred with the register files in the computing units 22 through 25.

Page 38: Amend the paragraph starting at line 14 and ending on page 39, line 16, as follows:

The following describes details of the instruction expansion circuit 2 for implementing the above-mentioned method. FIG.7 is a block diagram illustrating the instruction expansion circuit 2 in detail. With reference to FIG. 7, the circuit blocks and signal lines similar to those previously described with FIG. 1 are denoted by the same reference numerals. Reference numeral 40 denotes an instruction buffer for latching a compressed instruction code (32 bytes) from the data bus 4, reference numeral 41 (41a through 41h) denotes a compressed field bus having a constitution of 4-byte (one field or one header) x 8, reference numeral 42 denotes a field controller for analyzing header information, reference numeral 43 denotes a write enable bus constituted by eight signals for ~~enable~~ enabling a write operation on a 4-byte basis, reference numeral 44 denotes a field select signal for sorting fields, reference numerals 45a through 45h denote selectors for selecting one of eight 4-byte signals of the compressed field bus 41, reference numeral 67 denotes eight field signal lines indicating presence/absence of the fields 0 through 7, reference numeral 49 denotes an expansion field bus after field sorting, reference numeral 46 denotes an SIMD controller for controlling a field copy operation of each instruction in the SIMD mode, reference numeral 47 denotes a SIMD select signal line for controlling selection of a field to

be copied, and reference numeral 48 denotes a dual selector for selecting any two fields (IFG field and INT field).

Page 40: Amend the first full paragraph starting at line 6 as follows:

Because the instruction buffer 40 is 32 bytes long, it cannot hold a longest instruction ~~composes~~ composed of the header and the eight fields at a time. Therefore, in this case, two fetch operations are required. The field controller 42 outputs information indicating the second fetch to a refetch signal line 13. This signal is sent to the instruction registers 14 through 21 shown in FIG. 1. To be outputted in a refetch cycle is only the information of the field 7, so that, in this cycle, only the instruction register 21 corresponding to the field 7 latches (updates) the field data.

Page 41: Amend the paragraph starting at line 17 and ending on page 42, line 4, as follows:

Likewise, information for selecting the copy source fields for the fields 2 and 3, the fields 4 and 5, and the fields 6 and 7 is outputted to the corresponding dual selectors 48b through d. Each of the dual selectors 48b through d select two of the eight fields in the expanded field bus 49 and outputs the field 2 to the field bus 7, the field 3 to the field bus 8, the field 4 to the field bus 9, the field 5 to the field bus 10, the field 6 to the field bus 11, and the field 7 to the field bus 12. Thus, in the SIMD mode, according to the information specified in each instruction, the content of a particular field can be copied to another field. This allows ~~may~~ many computational operations with a small number of instructions.

Page 58: Amend the paragraph starting at line 23 and ending on page 59, line 2, as follows:

To be more specific, the address controller 61 outputs address 136 obtained by adding 28 to 108 of the instruction address bus 64 to the address bus 3. At transition from T6 to T7, the header address is not added with 1 and hence left at 1.

Page 67: Amend the last full paragraph as follows:

In the figure, reference numeral 110 denotes a program counter holding 32-bit address instruction address information, reference numeral 111 denotes a 32-bit adder, reference numeral 112 denotes a 2-bit header counter register, reference numeral 113 denotes a 2-bit adder, reference numeral 114 denotes a 2-input NOR circuit for outputting negate information of OR operation, reference numeral 115 denotes a selector for selecting address 32 or 28, reference numeral 115 denotes a selector for selecting the output of the selector 115 or 0, and reference numeral 117 denotes a 32-bit adder.

Page 75: Amend the paragraph starting at line 25 and ending on page 76, line 7, as follows:

The signal generator 145 operates basically in the same manner as the signal generator 144. However, because the select information of the fields 2 and 3 is outputted, no cop instruction is sent from the enable analyzer 141, so that the input corresponding thereto is fixed to 0 and select information 1 is outputted to make fields 2 and 3 be selected when no copy instruction (namely, not in the SIMD mode) is detected.

Page 76: Amend the first and second full paragraph starting on line 8, as follows:

Likewise, the signal generator 146 operates basically in the same manner as the signal generator 144. However, because the select information of the fields 4 and 5 is outputted, no copy instruction is sent from the enable analyzer 142, so that the input corresponding thereto is fixed to 0 and selected information 2 is outputted to make fields 4 and 5 be selected when no copy instruction (namely, not in the SIMD mode) is detected.

Further, the signal generator 147 operates basically in the same manner as the signal generator 144. However, because the select information of the fields 6 and 7 is outputted, no copy instruction is sent from the enable analyzer 143, so that the input corresponding thereto

is fixed to 0 select information 3 is outputted to make fields 2 and 3 be selected when no copy instruction (namely, not int the SIMD mode) is detected.

Page 79: Amend the first full paragraph starting on line 10 as follows:

FIG. 18 shows an instruction format of the present embodiment. In the figure, bits 0 through 27 of the INT field and the IFG field are the same as those of the first embodiment. Bits 28 and 29 of the IFG field indicate the address of the field. The IFG field is one of the fields 0, 2, 4 and 6 and bit assignment is as shown in the figure. Bit 30 (sync) of the IFG field indicates a synchronous signal. By inverting the sync bit for every instruction, the distinction between ~~instruction~~ instructions can be recognized. In the figure, it is specified that he sync bit becomes 0 for an even-number instruction and 1 for an odd-number instruction. Based on such an instruction format, detailed operations of the instruction expansion circuit 200, key to the present embodiment, will be described.

Page 82: Amend the last paragraph starting on line 16 and ending on page 83, line 11, as follows:

The select signal generator 211 receives information of the write enable bus 43, 41a, c, e, and g of "SIMD" and "S mode" and address information from the compressed field bus 41. From these pieces of information, the select signal generator outputs four bits of positional information (information indicating one of the four bits 41a, c, e, and h) of the field 0 to the select information line 206. If the field 0 is NOP-compressed, all four bits go 0. This ~~is~~, at the same time, provides the select information of the field 1 (information indicating one of the four bits 41b, d, f, and g). Likewise, the select signal generator outputs four bits of positional information (information indicating one of 41a, c, e, and h) of the field 2 to the select information line 207, four bits of positional information (information indicating one of 41a, c, e, and h) of the field 4 to the select information line 208, and four bits of positional information (information indicating one of 41a, c, e, and h) of the field 6 to the select

information line 209. The following describes detailed operations of the synchronizer 210 and the select signal generator 211.

Page 88: Amend the paragraph starting on line 25 and ending on page 89, line13, as follows:

Thus, the destination signal generation circuit 230 determines to which instruction field the 41a corresponds and determines the copy destination field in the SIMD mode, and then specifies the destination field of the 41a to the dual selector 202 via the select information line 206. Likewise, by determining the copy destination field in the SIMD mode, the destination signal generation circuit 231 determines to which instruction field the 42c corresponds, the destination signal generation circuit 232 determines to which instruction field the 41e corresponds, and the destination signal generation circuit 233 determines to which instruction field the 41 g corresponds and ~~specify~~ specifies the destination field numbers of the field information of the 41c, 41e, and 41g to the dual selectors.

Page 93: Amend the two paragraphs starting on line 22 and ending on page 94, line 19, as follows:

It should be noted that the instruction formats used in the above-mentioned embodiments are for illustrative purposes only and hence other formats may be used.

The above-mentioned embodiments 1, 2, and 3 are applied to a VLIW processor that presupposes static scheduling, but the present invention is not necessarily limited thereto. For example, the present invention is also applicable to a superscalar processor that performs dynamic scheduling. ~~On~~ An instruction in the superscalar processor is basically constituted by a fixed length of one field as described in the preceding embodiment. Such a processor incorporates a plurality of computing units and ~~an~~ instruction queues and has a dispatcher that checks the dependent relationship between the plurality of queued instructions and, if no dependency is found and a plurality of executable instructions are found, transfers these instructions to the plurality of computing units simultaneously. Therefore, as shown in FIG. 2

C14

of the present invention, if the SIMD mode is specified in the instruction format by "S-mode" and "SIMD," the above-mentioned dispatch unit transfers that instruction to the plurality of computing units, easily implementing the SIMD mode of the superscalar processor.

Page 95: Amend the first paragraph as follows:

C15

FIG. 28 shows the instruction ~~formed~~ format used in the present embodiment. As shown the IFG field and the INT field constitute one instruction each. Therefore, because it needs to be known to which format an instruction belongs, the IFG format is indicated when bit 31 is 0 and the INT format is indicated when bit 31 is 1. The subsequent bits have the same means as those of FIG. 24. However, "destination," "source 0," "source 1," and the spare block in the INT instruction have different bit positions. Therefore, the instruction simultaneously executable and inputs the extracted instructions into the computing units.